Advanced Enterprise Objects Frameworks

Session 712

Advanced Enterprise Objects Frameworks

Steve Miner, Ben Trumbull
WebObjects Engineers

Introduction

- Model recommendations
- Designing EOs
- Single Table Inheritance
- Shared Editing Contexts
- Delegate methods
- Data Synchronization
- Raw Rows and SQL

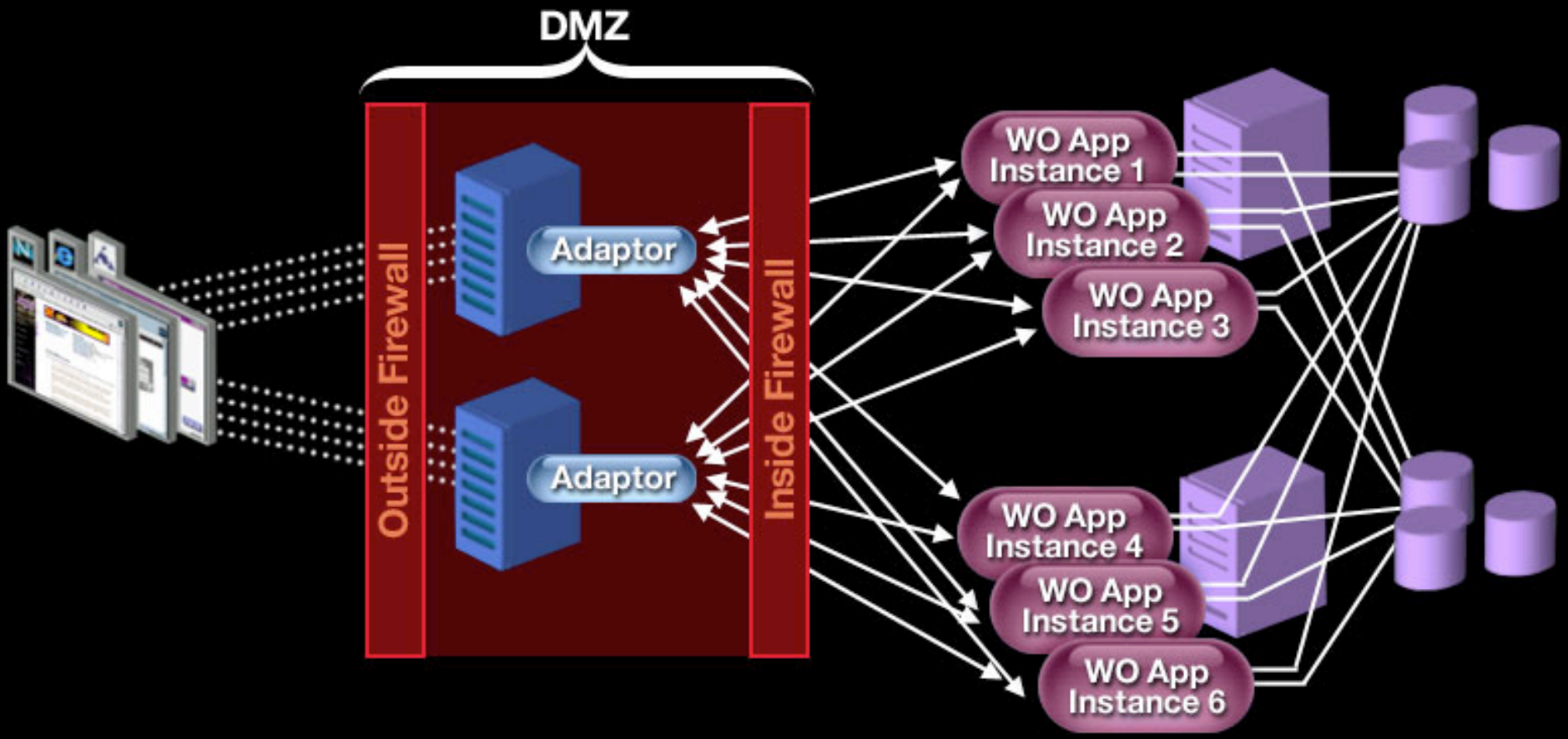
Last Year's Talk

- Eric Noyau's BugTracker example
- Separation of UI and "controller" logic
- Built-in testing during development
- **http://homepage.mac.com/eof/**

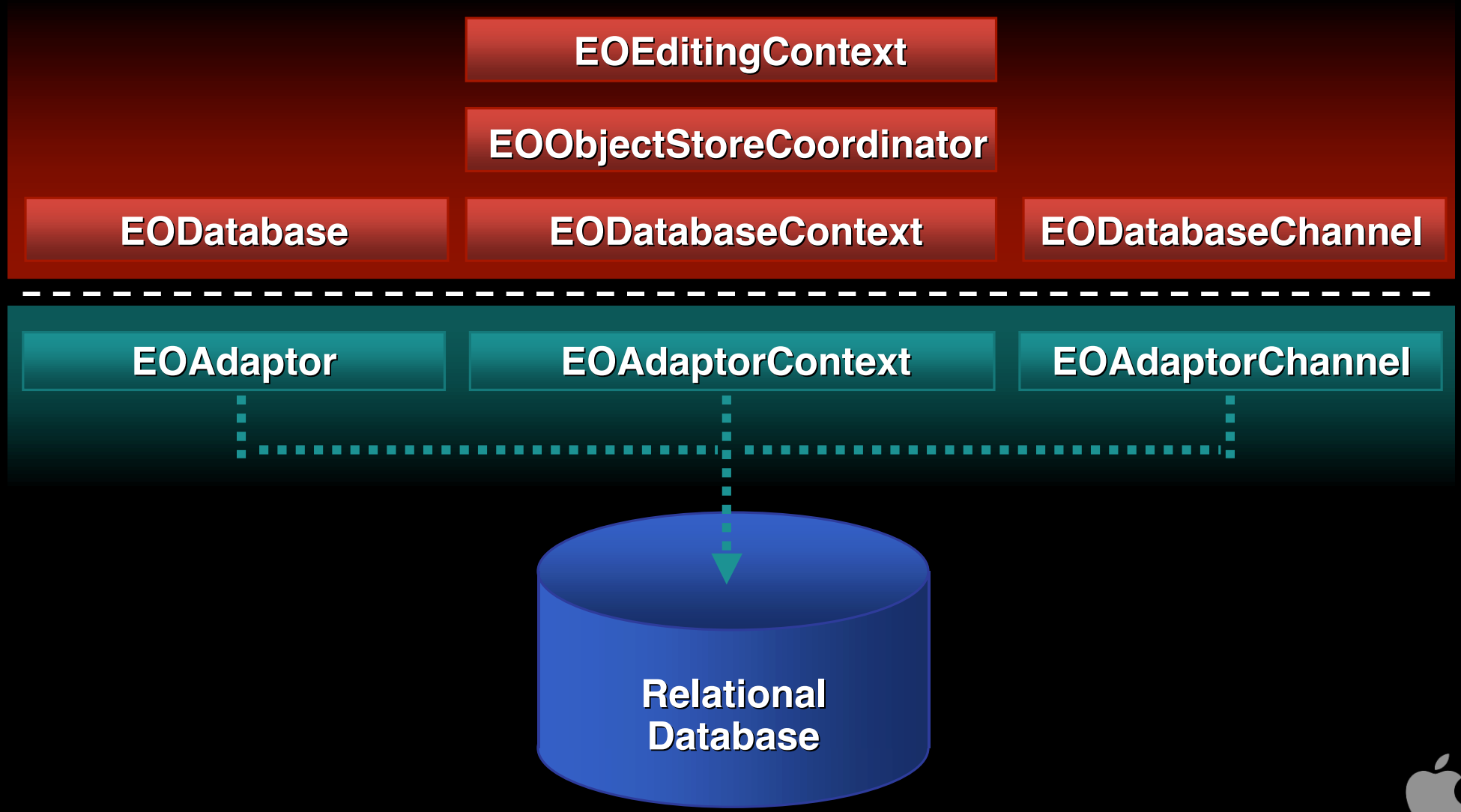
Technology Frameworks

- JavaFoundation
- JavaEOControl
- JavaEOAccess
- JavaJDBCAdaptor

WebObjects


DMZ
Outside Firewall
Adaptor
Adaptor
Inside Firewall
WO App Instance 1
WO App Instance 2
WO App Instance 3
WO App Instance 4
WO App Instance 5
WO App Instance 6
Browsers
Multiple HTTP Servers
Multiple WO App Servers
Multiple Databases

EOF


EOEditingContext
EOObjectStoreCoordinator
EODatabase
EODatabaseContext
EODatabaseChannel
EOAdaptor
EOAdaptorContext
EOAdaptorChannel
Relational Database

Model Recommendations

- Primary key
  - Simple integer
  - Usually not a foreign key
- Class properties
  - Be careful with large to-many
- Blobs
  - Separate tables
  - Not used for locking

Primary Key Generation

- Automatic for integer PKs
- databaseContextNewPrimaryKey() delegate
- awakeFromInsertion()

Designing EOs

- EOEnterpriseObject interface
- EOCustomObject class
  - Basic EO
    willRead() in getters
    willChange() in setters
- EOGenericRecord class
  - No code required
  - Automatic storage
  - Deferred faulting
  - subclassable

Keep EO Code Clean

- Use Foundation and EOControl
- Maybe EOUtilties (EOAccess)
- Avoid DB specific code
- No WebObjects Framework code

KeyValueCoding

- Interfaces
    - NSKeyValueCoding
    - NSKeyValueCodingAdditions
    - EOKeyValueCoding
    - EOKeyValueCodingAdditions
- Utility static methods—any object
- DefaultImplementation

KVC Basics

- valueForKey(key)
- takeValueForKey(value, key)
- valueForKeyPath(keypath)
- takeValueForKeyPath(value, keypath)
- Also implemented by NSDictionary
- NSArray @ count, @min, @max, @sum, @avg

KVC (Cont.)

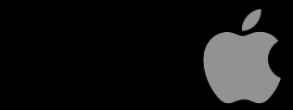
- storedValueForKey(key)
- takeStoredValueForKey(value, key)
- valuesForKeys(keys)
- takeValuesFromDictionary(dict)
- public static boolean canAccessFieldsDirectly()
- public static boolean **shouldUseStoredAccessors()**

valueForKey(“name”)

- public getName(), name(), isName()
- underbar _getName(), _name(), _isName()
- field _name, _isName, name, isName

storedValueForKey (“name”)

- underbar _getName(), _name(), _isName()
- field _name, _isName, name, isName
- public getName(), name(), isName()



takeValueForKey (val, “name”)

- public setName()
- underbar _setName()
- field _name, _isName, name, isName

takeStoredValueForKey (val, “name”)

- underbar _setName()
- field _name, _isName, name, isName
- public setName()

NSValidation

- validateValueForKey
    - Calls custom validateKEY
- Utility—any object
- DefaultImplementation

Validation Example

```
public String validateName(String name)
      throws NSValidation.ValidationException {
   if (name == null || name.length() < 5)
      throw new NSValidation.ValidationException(
   “The name should contain more than 4 characters”,
   this, “name”);
   return name;
}
```

EOValidation

- validateForInsert
- validateForUpdate
- validateForSave
- validateForDelete

Other EO methods

- handleQueryWithUnboundKey()
- handleTakeValueForUnboundKey()
- unableToSetNullForKey()
- awakeFromInsertion()
- awakeFromFetch()

Don’t Override

- willChange()
- willRead()
- willReadRelationship()
- storedValueForKey()
- takeStoredValueForKey()

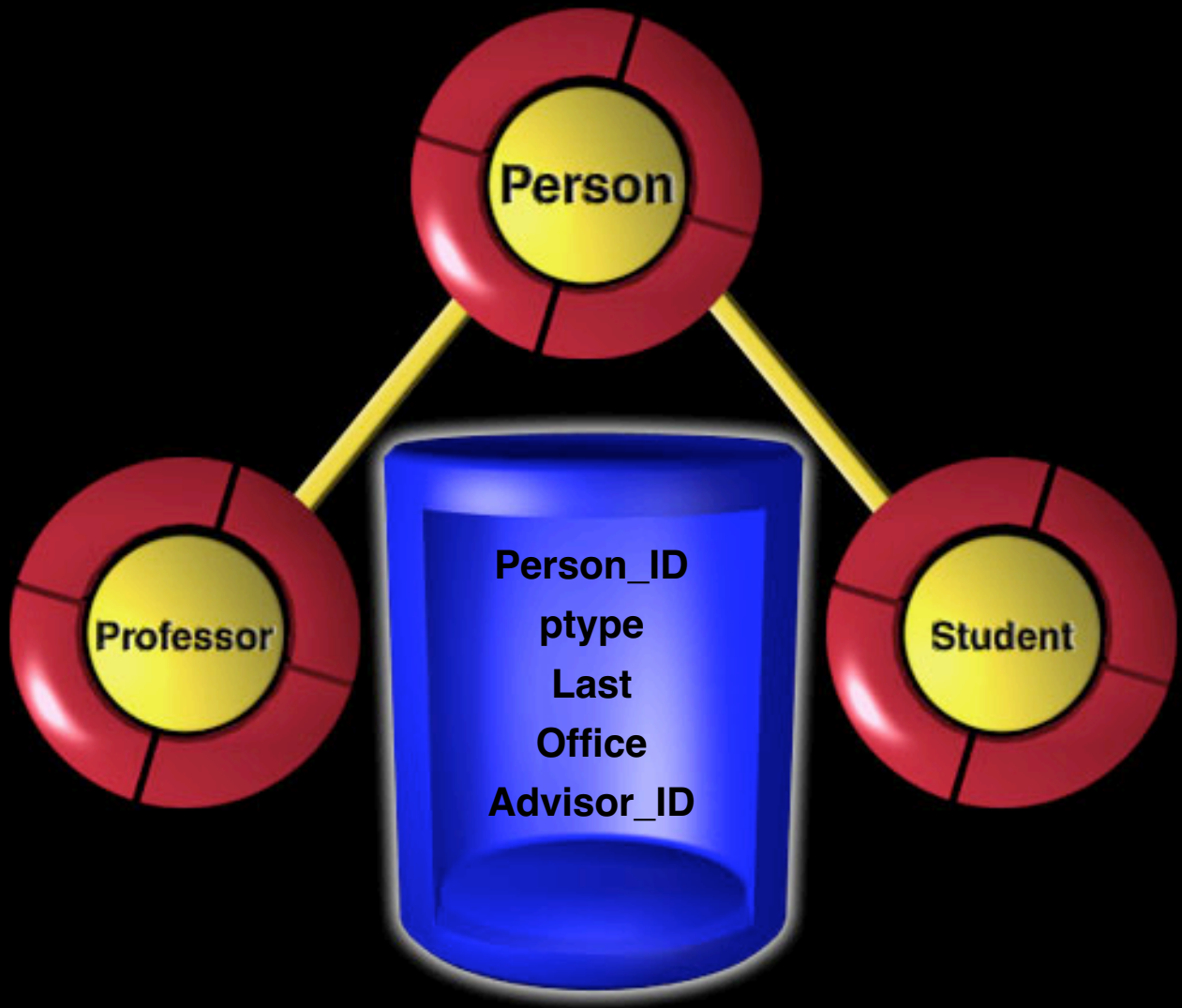
Inheritance

- Vertical
    - Natural for OOP, simple maintenance
    - Expensive fetches (joins)
- Horizontal
    - Fetch without joins
    - More maintenance overhead

Single Table Inheritance

- Subentities map different columns
- Must allow null
- Single fetch for deep inheritance
- Avoids limitations on sorting

Single Table Inheritance



Single Table Optimization

- Subentity type attribute: sub
- Restricting qualifier: sub = N
- Root entity awakeFromInsertion()
    - setSub (mySub())
- Unique mySub() for each subentity
- Never change the sub attribute

Shared Editing Contexts

- Shared objects set in model
- Common EOs (catalog)
- No outgoing relationships

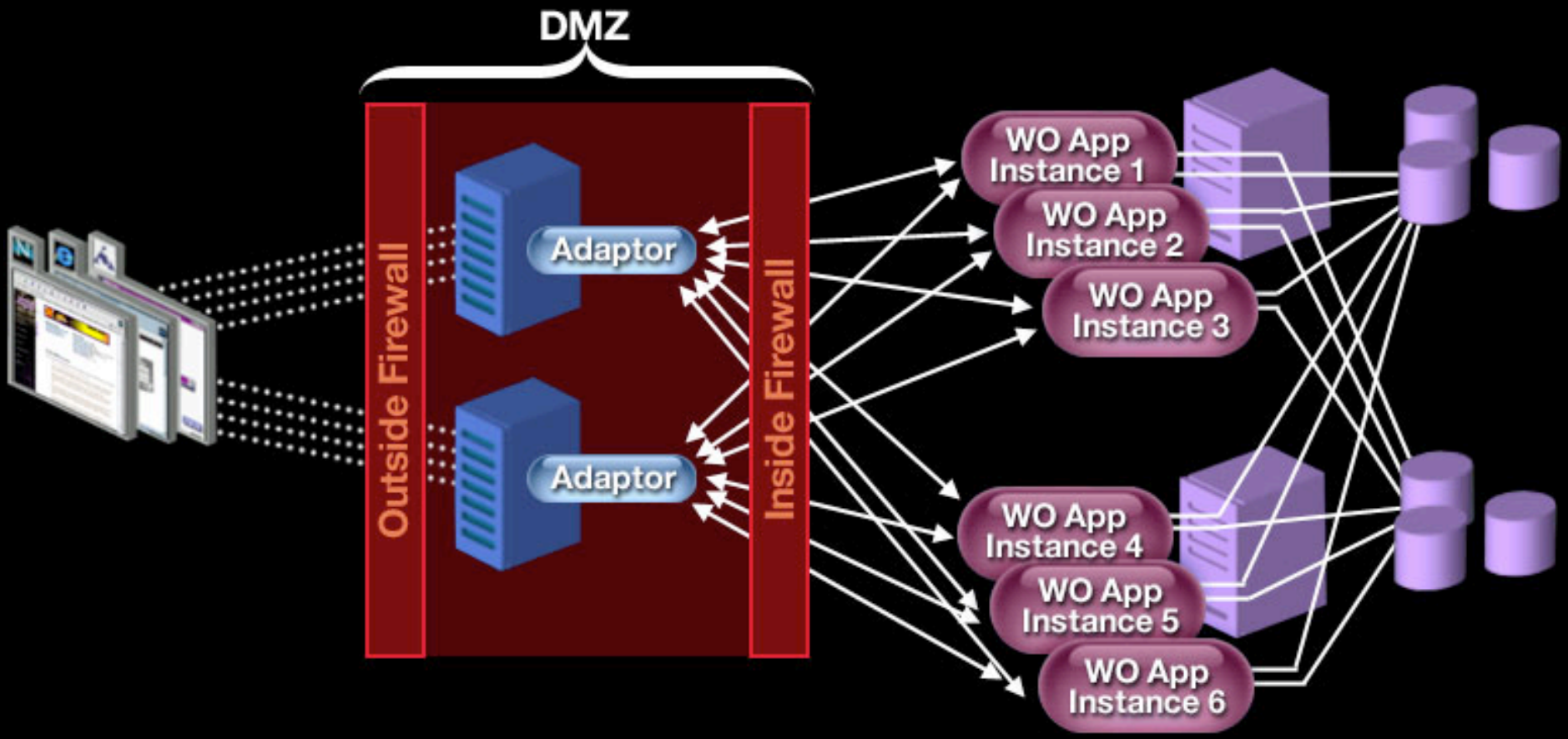
Delegate methods

- EOControl
  - EOEditingContext
- EOAccess
  - EODatabaseContext
  - EOAdaptor
  - EOAdaptorContext
  - setDefaultDelegate

Data Synchronization

- EOF is a big cache
- Multiple WOA instances
- External writers
- Optimistic locking exception

WebObjects

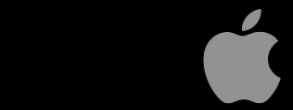

DMZ
Outside Firewall
Adaptor
Adaptor
Inside Firewall
WO App Instance 1
WO App Instance 2
WO App Instance 3
WO App Instance 4
WO App Instance 5
WO App Instance 6
Browsers
Multiple HTTP Servers
Multiple WO App Servers
Multiple Databases

Approaches

- Do nothing, runtime exception
- Avoidance
- Recovery

Avoidance

- refaultObjects() and setFetchTimestamp()
- setRefreshesRefetchedObjects()
- invalidateObjectsWithGlobalIDs()
- Code to mimic triggers



Change Notification Framework

- David Neumann (Apple)
- WireHose improved version
- http://www.wirehose.com/download/

Recovery

- Catch EOGeneralAdaptorException
  - userInfo (constants in EOAdaptorChannel)
  - AdaptorFailureKey == AdaptorOptimisticLockingFailure
  - FailedAdaptorOperationKey

Demo

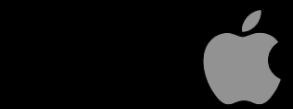
Ben Trumbull
WebObjects Engineering

Raw Rows

- Fetching raw row dictionaries
- faultForRawRow() needs PK
- Declaring a KVC Variable
    - ThinkMovies example
    - Mix EOs and raw rows
    - /** @TypeInfo Movie */
      public NSKeyValueCoding aMovie;

Using SQL

- Custom SQL query in fetch spec hints
- SQL expressions for attribute definitions
- EOEntity external query
- EOUtilities.rawRowsForSQL()
- EOAdaptorChannel. evaluateExpression()
- Custom EOQualifier



Demo

Ben Trumbull
WebObjects Engineering

Summary

- Model recommendations
- Designing EOs
- Single Table Inheritance
- Shared Editing Contexts
- Delegate methods
- Data Synchronization
- Raw Rows and SQL

WebObjects Beta

- To be considered for the beta

  Appleseed.apple.com/webobjects

WebObjects Lab

- Located downstairs in Room L
- Lab hours
  - Monday 12:00pm–6:00pm
  - Tuesday 9:00am–2:00pm
  - Wednesday 9:00am–6:00pm
  - Thursday 9:00am–6:00pm
  - Friday 9:00am–6:00pm

Roadmap

FF013 WebObjects

Room A1
Fri., 3:30pm

Who to Contact

Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

Bob Fraser
WebObjects Product Manager
webobjects@apple.com

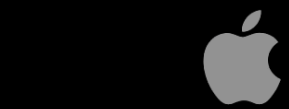
Apple Professional Services (Training, Support, Consulting)
(800) 848-6398
services@apple.com

http://developer.apple.com/wwdc2002/urls.html

For More Information

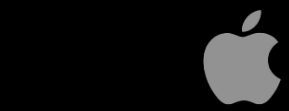
- WebObjects Developer Documentation
  http://developer.apple.com/techpubs/webobjects
- Apple Professional Services Technical Support
  (www.apple.com/services/technicalsupport)
- Other places
  - **www.apple.com/webobjects**
  - **developer.apple.com/webobjects**
  - **www.apple.com/services**
  - **www.info.apple.com/webobjects**

Subscribe to:
webobjects-announce@apple.com

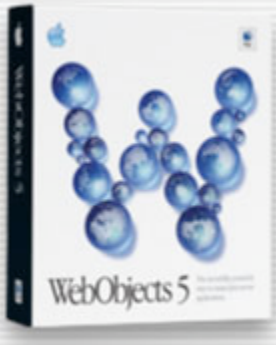


How to Access Documentation

- Most up-to-date: PDF and HTML
  http://developer.apple.com/techpubs/webobjects
- Hardcopy print-on-demand
  Vervante.com under Related Resources
- Product CD
  Documents folder and installed in /Developer/Documentation/WebObjects
- In the box (localized)
  Installation Guides, What's New, WebObjects Overview, Java Client Desktop Applications, Discovering WebObjects for HTML
- Check ADC News for latest updates
  http://developer.apple.com/devnews



Q&A



Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

http://developer.apple.com/wwdc2002/urls.html


WWDC2002


WWDC2002


WWDC2002